

Parte 1 – Classificação de um Padrão de Voz

1. Correr o Ficheiro “Baseline”

Inicialmente corremos o script Baseline fornecido pelo professor nas aulas de laboratório. Colocámos todas as variáveis dos STEP[N] com N de 0 a 4 com valor positivo (true). Obtivemos uma percentagem de 80,3653% (528/657) de ficheiros corretamente classificados usando os comandos “-t 0” que corresponde a um Kernel linear. A maneira como isto se processou foi do seguinte modo, o nosso sistema retirou inicialmente alguns “features” dos ficheiros de áudio que nos eram fornecidos e criou um modelo utilizando a biblioteca de Support Vector Machines. Para que o sistema conseguisse aprender por si, foi-lhe fornecido um conjunto de ficheiros de treino, e este foi validado no conjunto de ficheiros de um ambiente de desenvolvimento. Conseguimos comprovar a qualidade do nosso classificador pois foram-nos fornecidas labels para cada um dos ficheiros de treino e desenvolvimento.

A avaliação do nosso sistema será feita nos ficheiros de teste, com as labels que apenas os professores possuem.

2. Melhorar o Ficheiro “Baseline”

Inicialmente o grupo decidiu alterar os parâmetros do Passo 3 que corresponde a treinar o nosso modelo de Support Vector Machine. Para tal, fomos ler os parâmetros que podiam ser alterados e corremos cada um deles individualmente:

- -t 1 (Polynomial)
- -t 2 (Radial Basis)
- -t 3 (Sigmoid)
- -s 0 C-SVC
- -s 1 nu-SVC
- -s 2 one-class SVM
- -s 3 epsilon-SVR
- -s 4 nu-SVR

Em todas estas modificações o resultado foi idêntico com uma taxa de sucesso na identificação dos ficheiros do ambiente de desenvolvimento de cerca de 76,1035% (500/657). O que é ligeiramente inferior aos resultados anteriores que consideram um Kernel linear, no entanto conseguimos reparar que todos estes realizavam muito menos iterações do que o Kernel linear. Tal se deve ao facto de termos poucos ficheiros para treinar o nosso sistema e para além disso cada um tem muitos parâmetros, features correspondentes.

De modo a aumentar um pouco a qualidade do classificador, e considerando desta vez o Kernel Linear por ter tido melhores resultados, decidimos substituir o método de extração de features pelo método Extended Gemaps.

O grupo decidiu então mudar de abordagem e não utilizar a Baseline inicialmente dada. Para isso vamos tentar explicar a linha de raciocínio:

- Audio (.wav) → Feature Extraction (using MFCC)
- Features obtidos
- Treino do modelo (Neural Network)
- Obtenção das classes
- Utilização de PCA (Principal component Analysis)
- Classificação da decisão

Para isto foram consideradas bibliotecas já existentes em 2 linguagens de programação (Python e Matlab) para realizar os objetivos anteriores.

3. Decisões e Comentários Finais